

Building and using a custom CLI device pack and YANG module for building KPIs

This Document provides following high level steps

1. Building CLI Device package
2. Building YANG Model
3. Packaging them for Import into Crosswork and Data Gateway (CDG)
4. Installing / Updating Crosswork and Data Gateway (CDG)
5. Using CLI YANG models on Crosswork HI

1. Building CLI Device package

A CLI device pack by default emits the data in XML format, but CLI collector in CDG has a feature wherein it can convert this XML data into GPBKV telemetry data using a YANG model.

The pre-requisites of such conversion are:

1. A YANG model representing the data collected by the CLI device package.
2. The device pack's action RPL XSD(XML Schema Definition) must match with YANG model and needs to be augmented with it
3. Custom CLI device pack and YANG model should have same file name, barring the file extension.
4. YANG model and device pack must be available as system packages in CDG.
5. YANG model must be available as custom model in CW.

Currently (CDG <= v1.1.3) has restrictions on ability to upload a device pack along with the YANG model that can be used to format the XML data output. However existing, pre-installed system device packs such as cli_xr_generic_show_command, cli_reachability, etc. support YANGification of collected data. This method details the steps to overwrite and update stock device package and yang models.

ATTENTION: These steps would need updating system / stock package and restarting services and may have service impact

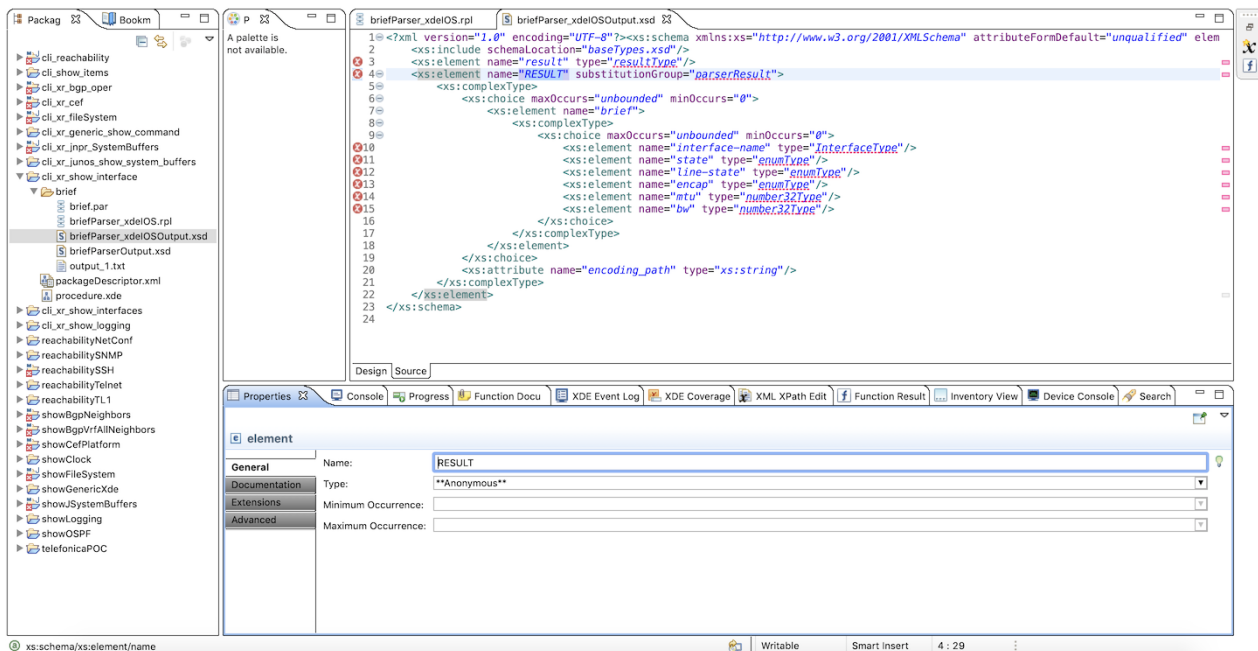
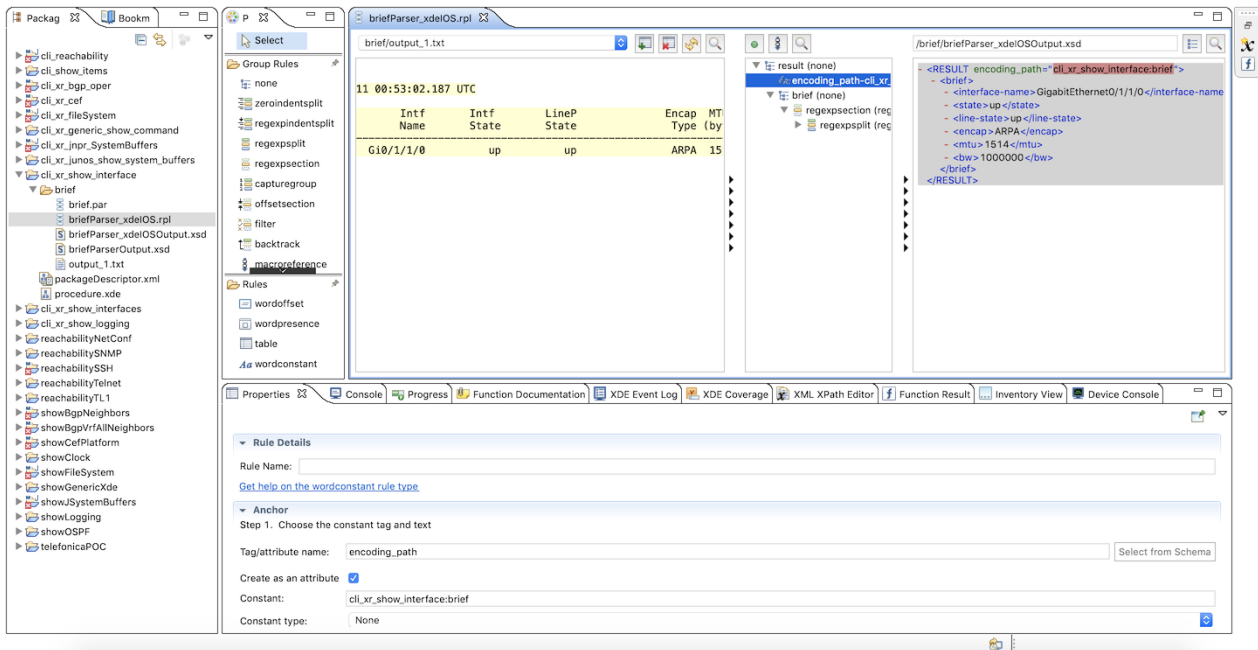
Use XDE/PAL Eclipse IDE to build device package

- **Create / Modify the XDE/PAL device pack, such that the root `RESULT` has an attribute `encoding_path` , with the value of the gather path from YANG module.**

[See example CLI device pack archive](#)

The attribute `encoding_path="cli_xr_show_interface:brief"` , where `cli_xr_show_interface` is the YANG module name, and `brief` is the top-level container in model.

The element immediately under `RESULT` root is `brief`` . Within the `brief` element all child elements correspond leaves from YANG module.



- Export the device pack as XDE/PAL archive file.

Both the files should have same name, but different extensions, i.e. .xar and .yang for device pack and YANG model respectively.

The file name must have cli_xr_ as prefix for Crosswork to consider it for CLI collection in 3.x releases

2. Building YANG Model

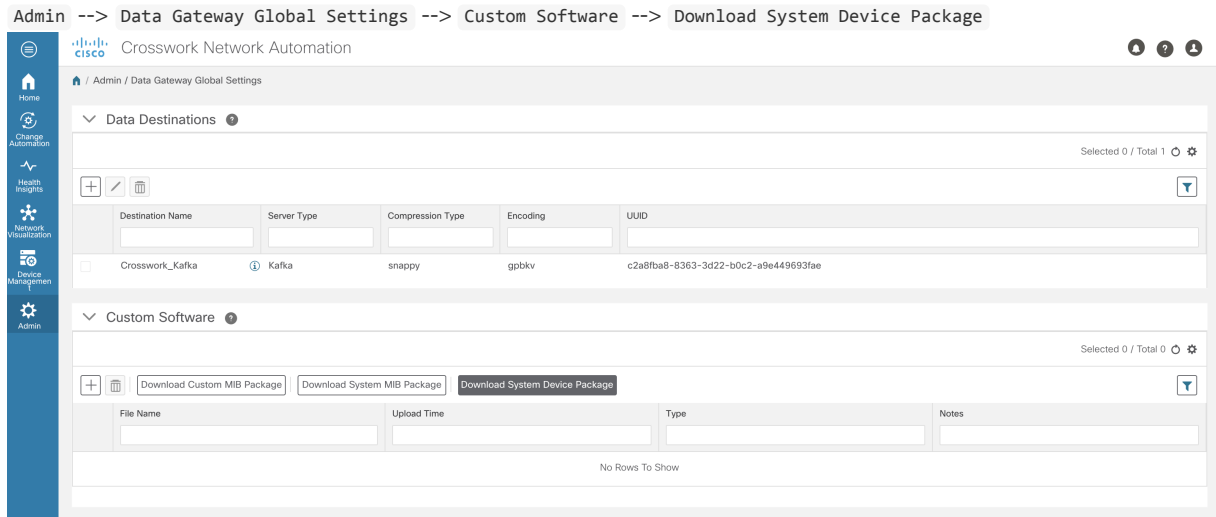
- Create a YANG model as required for the given device pack output.

See example YANG model [here](#)

3. Packaging them for Import into Crosswork and Data Gateway (CDG)

YANG model and device pack must be made available as system packages in CDG. Following steps are to export and modify system packages

- Download the existing system CLI device packages from Crosswork UI.



Update the system archive with the newly created artifacts.

```
# extract contents
tar xzvf system-cli-device-packages.tar.gz

# copy artifacts
cp cli_xr_show_interface.yang yang/
cp cli_xr_show_interface.xar xar/

# re-package
tar -czvf system-cli-device-packages.tar.gz yang/ xar/
```

- Prepare YANG model for Crosswork/HealthInsights

YANG model must be made available as a custom model in CW. The model needs to be uploaded as a third-party/custom MIB+YANG package.

Follow the [documentation here](#) for packaging the YANG model, there is no need for MIB files in this case.

Using following directory format add the new .yang file to create a tar.gz archive that also includes a custom_seedlist plain text file with name of the yang model

```
custom-mib-packages/
custom-mib-packages/yang/
custom-mib-packages/yang/cli_xr_show_interface.yang
custom-mib-packages/mib/
custom-mib-packages/custom_seedlist
```

Create Tar archive

```
tar -cJvf custom-mib-package.tar.xz custom-mib-packages
```

4. Installing / Updating Crosswork and Data Gateway (CDG)

system-cli-device-packages.tar.gz and custom_yang.tar.xz files created in Step 3, will be used to import into Crosswork and CDG

- Upload device package into Crosswork using SCP

```
scp system-cli-device-packages.tar.gz cw-admin@<CW-IP>:/home/cw-admin
```

- Log in to Crosswork VM shell, and move the archive to a dir where CDG expects it

```
mv system-cli-device-packages.tar.gz /mnt/cw_dgmgrfs/repo/system-device-packages/
```

[See example system pack archive](#)

- Login to CDG VM and use interactive menu to restart the Collectors and Reboot VM

```
5 Troubleshooting --> 6 Remove All Collectors and Reboot VM
```

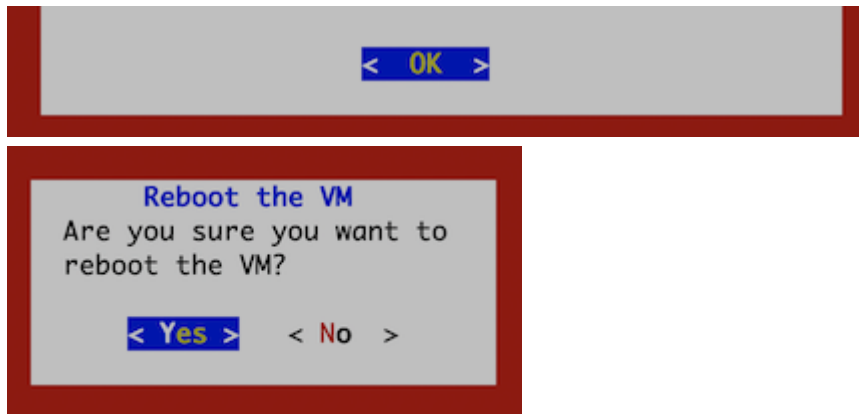
Main Menu - Please Choose an Option:

- 1 Export Enrollment Package
- 2 Show System Settings
- 3 Change Current System Settings
- 4 Vitals
- 5 Troubleshooting**
- p Change Passphrase
- l Logout

< OK >

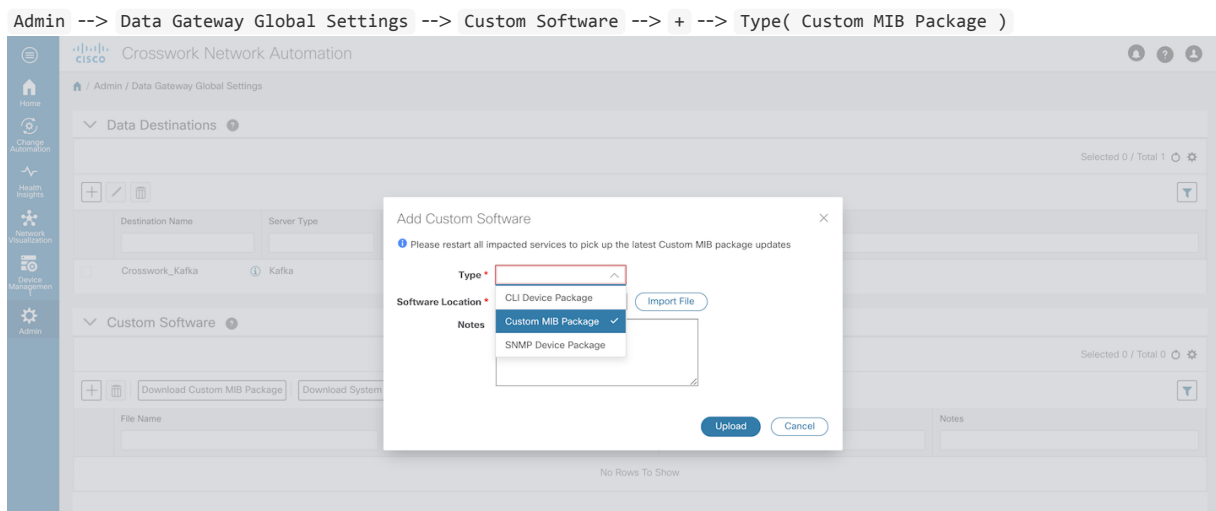
Troubleshooting - Please Choose an Option:

- 1 Ping a Host
- 2 Traceroute to a Host
- 3 NTP Status
- 4 System Uptime
- 5 Run show-tech
- 6 Remove All Collectors and Reboot VM
- 7 Reboot VM**
- x Exit Menu



- **Upload custom yang archive into Crosswork**

- Import the archive using Crosswork UI
- [See example archive](#)



After the above steps, Crosswork collection services can now be used to request the data collected by the custom device pack as per the YANG model.

To do so any Crosswork app (like HI) can request to collect data using SensorData of type CliYangSensor.

The sensor data path is a combination of YANG gather path and some optional command which can be sent to the CLI device pack for runtime values to be used by its procedure.

- **Restart Crosswork components**

- Once the Upload is successful, use Crosswork UI `Admin --> Crosswork Manager` and restart following services `Health Insights` , `robot-fleet` and `robot-alerting`

5. Using CLI YANG models on Crosswork HI

- Now Helath Insights can be used to create a new KPI using KPI wizard and selecting the new CLI based yang model. Also CDG is has the required Device packages to start the CLI collection and conversion to kvjson etc. Use [Health Insights Developer Guide](#) for creating custom KPIs
- To debug issues, look for if the subscription is successful in Collection service UI for errors and counters about the collection.
If collection is happening but still KPI dashbaord does not show desired data , validate that the influxDB has the data in right way.
Also validate `robot-astack-pipeline` stats do not have any errors.